

## BAB 6

### KESIMPULAN DAN SARAN

#### 6.1. Kesimpulan

- a. Perancangan purwarupa AGV terintegrasi menggunakan metode rasional, mampu menunjukkan prioritas perancangan menghasilkan rancangan purwarupa AGV dengan tipe *Industrial Reach Forklift Truck* yang menggunakan basis pergerakan perunut garis dan integrasi server dengan basis RF.
- b. Purwarupa AGV berhasil melakukan pengujian tugas yang diberikan dengan presentase keberhasilan yaitu 94% pada proses penyimpanan, 92.67% pada proses pengambilan, 96.36% pada proses kembali ke *docking station*, dan 100% respon *feedback*. Purwarupa AGV mampu melaksanakan sebagian besar perintah tugas yang diberikan, maka layak digunakan sebagai alat peraga pada pembelajaran di Laboratorium Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.

#### 6.2. Saran

Saran pengembangan untuk penelitian ini adalah:

- a. Penggunaan metode PID untuk pergerakan bisa dikembangkan ke model PI, PD, atau PID, untuk jalur dan pergerakan yang lebih rumit.
- b. Pembuatan *casing body* untuk tampilan agar semakin mirip dengan *forklift*.
- c. Penggunaan tipe baterai lain yang memiliki keluaran 5V dan arus yang stabil.
- d. Penambahan unit purwarupa AGV untuk *multitasking* dan pengembangan server.
- e. Penghitungan waktu tempuh antar jalur, bisa digunakan untuk perbandingan rute terpendek dan terefektif apabila ada 2 atau lebih purwarupa AGV.

## DAFTAR PUSTAKA

- Andika, F. (2015). Perancangan dan Implementasi Sistem Kendali untuk Navigasi AGV, Fakultas Teknik Elektro, Universitas Gadjah Mada
- Arduino. (n.d). Arduino Nano ATmega328 datasheet. 26 Mei 2017. <https://www.arduino.cc/en/Guide/ArduinoNano> .
- Arif, M. (2016). Bahan Ajar Rancangan Teknik Industri, Deepublish, Indonesia.
- Baboli, A., Okamoto, J., Tsuzuki, M. S., Martins, T. C., Miyagi, P. E., & Junqueira, F. (2015). *Intelligent Manufacturing System Configuration and Optimization Considering Mobile Robots, Multi-Functional Machines and Human Operators: New Facilities and Challenge for Industrial Engineering*. IFAC-PapersOnLine, 48(3), 1912-1917.
- Cohen, L. (1995). *Quality Function Deployment : How Make QFD Work for You*. Addison – Wesley Publishing Company. Massachusetts.
- Cross, N. (1994). Engineering Design Methods, 2nd Ed. John Willey and Sons Inc. England.
- Gokhale, A. A. (1996). *Effectiveness of Computer Simulation for Enhancing Higher Order Thinking*.
- Groover, M.P., 2001, *Automation, Production System, and Computer –Integrated Manufacturing (2nd Edition)*. New Jersey: Prentice Hall International, Inc.
- Jaiganesh, V., Kumar, J. D., & Girijadevi, J. (2014). Automated guided vehicle with robotic logistics system. *Procedia Engineering*, 97, 2011-2021.
- Kilian, C. T. (2000). *Modern Control Technology*. Delmar Thomson Learning,.
- Martínez-Barberá, H., & Herrero-Pérez, D. (2010). *Autonomous navigation of an automated guided vehicle in industrial environments. Robotics and Computer-Integrated Manufacturing*, 26(4), 296-311.
- Richards, G. (2014). *Warehouse Management: A complete guide to improving efficiency and minimizing costs in the modern warehouse*. Kogan Page Publishers.
- SAP (2001). *Warehouse Management Guide, release 4.6C*, SAP AG.
- Sekaran, U. (2006). Metodologi Penelitian untuk Bisnis, Edisi 4, Buku 2, Salemba Empat, Indonesia.
- Masykuri, L. S. (2012) *Prototype Automatic Guided Vehicle (AGV)*. D3 thesis, UNY.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*(Vol. 1, No. 1). Cambridge: MIT press.

Syed Mohd Safwan, S. M. D. (2009) *Development of control system for automated guided vehicle (AGV). Faculty of Mechanical Engineering , University Malaysia Pahang.*

Ulrich, K. T. & Eppinger, S. D. (2008). *Product design and development 4th Edition.* New York: Mc Graw-Hill.

Ullrich, G. (2015). *Automated Guided Vehicle Systems: A Primer with Practical Applications.* Springer. London

Visioli, A. (2006). *Practical PID control.* Springer Science & Business Media.

Yunarto, H. I., & Santika, M. G. (2005). *Business Concept Implementation Series in Inventory Management.* Jakarta: PT Elex Media Komputindo.



**LAMPIRAN 1**  
**DATA NILAI LAPORAN DAN PRESENTASI TUGAS BESAR PRAKTIKUM SKI**  
**SEMESTER GASAL 2016/2017**

No.Mhs	Kel	Laporan					Presentasi TB
		Bab 1	Bab 2	Bab 3	Bab 4	Final	
7311	A1	75	70	75	65	72	55
7712		75	70	75	65	72	54
6444		0	0	0	0	0	0
7821	A2	75	60	65	70	67	54
7260		75	60	65	70	67	51
7908	A3	75	75	75	70	74	39
7438		75	75	75	70	74	48
7980		75	75	75	70	74	44
7740	A4	70	70	65	70	68	61
7224		70	70	65	70	68	64
7721	A5	75	70	70	70	71	60
7295		75	70	70	70	71	56
6814		75	70	70	70	71	51
7724	A6	60	65	70	0	53	70
6850		60	65	70	0	53	53
7697	A7	80	80	75	75	77	68
7751		80	80	75	75	77	64
6760		80	80	75	75	77	59
No.Mhs	Kel	Laporan					Presentasi TB
		Bab 1	Bab 2	Bab 3	Bab 4	Final	
7754	B1	75	60	60	65	64	64
7733		75	60	60	65	64	55
7289		75	60	60	65	64	56
7392	B2	60	60	60	60	60	62
7300		60	60	60	60	60	49
7758	B3	70	80	75	75	75	45
7832		70	80	75	75	75	46
7234		70	80	75	75	75	55
8035	B4	80	80	75	75	77	69
7554		80	80	75	75	77	53
7731	B5	70	70	65	65	67	54
7742		70	70	65	65	67	50
7835		70	70	65	65	67	39
8060	B6	80	80	80	75	79	70
8009		80	80	80	75	79	57
7532		80	80	80	75	79	55
7753	B7	75	70	70	70	71	59
7725		75	70	70	70	71	59

No.Mhs	Kel	Laporan					Presentasi TB
		Bab 1	Bab 2	Bab 3	Bab 4	Final	
7769	C1	60	60	60	60	60	64
7774		60	60	60	60	60	0
7780		60	60	60	60	60	65
8078	C2	70	70	70	70	70	55
6854		0	0	0	0	0	0
7752		70	70	70	70	70	53
7813	C3	70	70	70	70	70	59
7746		70	70	70	70	70	66
7694	C4	75	75	75	75	75	61
7786		75	75	75	75	75	63
7773		75	75	75	75	75	64
7795	C5	70	70	70	70	70	57
7788		70	70	70	70	70	61
7747	C6	60	60	60	60	60	0
7367		70	70	70	70	70	57
7760		70	70	70	70	70	47
7767	C7	70	70	70	70	70	55
7796		70	70	70	70	70	56

No.Mhs	Kel	Laporan					Presentasi TB
		Bab 1	Bab 2	Bab 3	Bab 4	Final	
7679	D1	75	75	75	75	75	71
6881		75	75	75	75	75	0
7784	D2	60	60	60	60	60	62
7791		60	60	60	60	60	59
7822	D3	75	75	75	75	75	77
8085		75	75	75	75	75	73
7671	D4	70	70	70	70	70	63
8076		70	70	70	70	70	63
0						0	
7765	D5	70	70	70	70	70	39
7071		70	70	70	70	70	50
8062	D6	75	75	75	75	75	80
7513		75	75	75	75	75	47

--	--	--	--	--	--	--	--

No.Mhs	Kel	Laporan					Presentasi TB
		Bab 1	Bab 2	Bab 3	Bab 4	Final	
6965	E1	65	65	65	65	65	48
7800		65	65	65	65	65	64
8032		65	65	65	65	65	63
7887	E2	65	65	65	65	65	64
7236		65	65	65	65	65	61
7759	E3	75	75	75	75	75	73
7068		0	0	0	0	0	0
7838	E4	60	60	60	60	60	63
7775		60	60	60	60	60	57

No.Mhs	Kel	Laporan					Presentasi TB
		Bab 1	Bab 2	Bab 3	Bab 4	Final	
6857	F1	90	75	80	75	80	56
7464		90	75	80	75	80	64
7830		90	75	80	75	80	64
7705	F2	75	75	75	80	76	70
7627		75	75	75	80	76	69
7719	F3	85	80	80	90	83	69
7904		85	80	80	90	83	68
7481		85	80	80	90	83	54
7645	F4	90	85	85	85	86	69
6858		0	0	0	0	0	0
7277		90	85	85	85	86	66
7781	F5	65	60	55	60	59	37
7561		65	60	55	60	59	36
7905	F6	90	90	80	80	84	67
7704		90	90	80	80	84	75
7231	F7	70	65	65	70	67	62
7670		70	65	65	70	67	60
7304		0	0	0	0	0	0







































Keterangan:





Tugas Besar membahas tentang sistem otomasi pada suatu rantai produksi. Mahasiswa mencari video atau dokumentasi mengenai suatu proses produksi di industri manufaktur sesuai tema yang ditentukan, dan mengambil satu proses produksi dan otomasi yang dilakukan. Mahasiswa menelaah komponen dan proses yang terjadi, kemudian menerjemahkannya ke dalam perancangan komponen kerja yang digunakan dan urutannya. Hasil analisa tersebut kemudian dipresentasikan.

Pada bab 1, mahasiswa membahas mengenai latar belakang proses otomasi yang terjadi. Pada bab 2, mahasiswa membahas mengenai teori yang mendasari cara kerja proses otomasi yang dibahas. Pada bab 3, mahasiswa menganalisa proses dan urutan kerja pada sistem otomasi yang dipilih. Pada bab 4 mahasiswa membahas mengenai komponen apa saja yang digunakan dan kecocokan penggunaannya terhadap satu sama lain. Hasil dari penulisan dan pembahasan ini kemudian dipresentasikan sebagai bukti tingkat pemahaman yang dimiliki oleh mahasiswa terhadap sistem otomasi yang dibahas.

## LAMPIRAN 2

### HOUSE OF QUALITY MATRIX

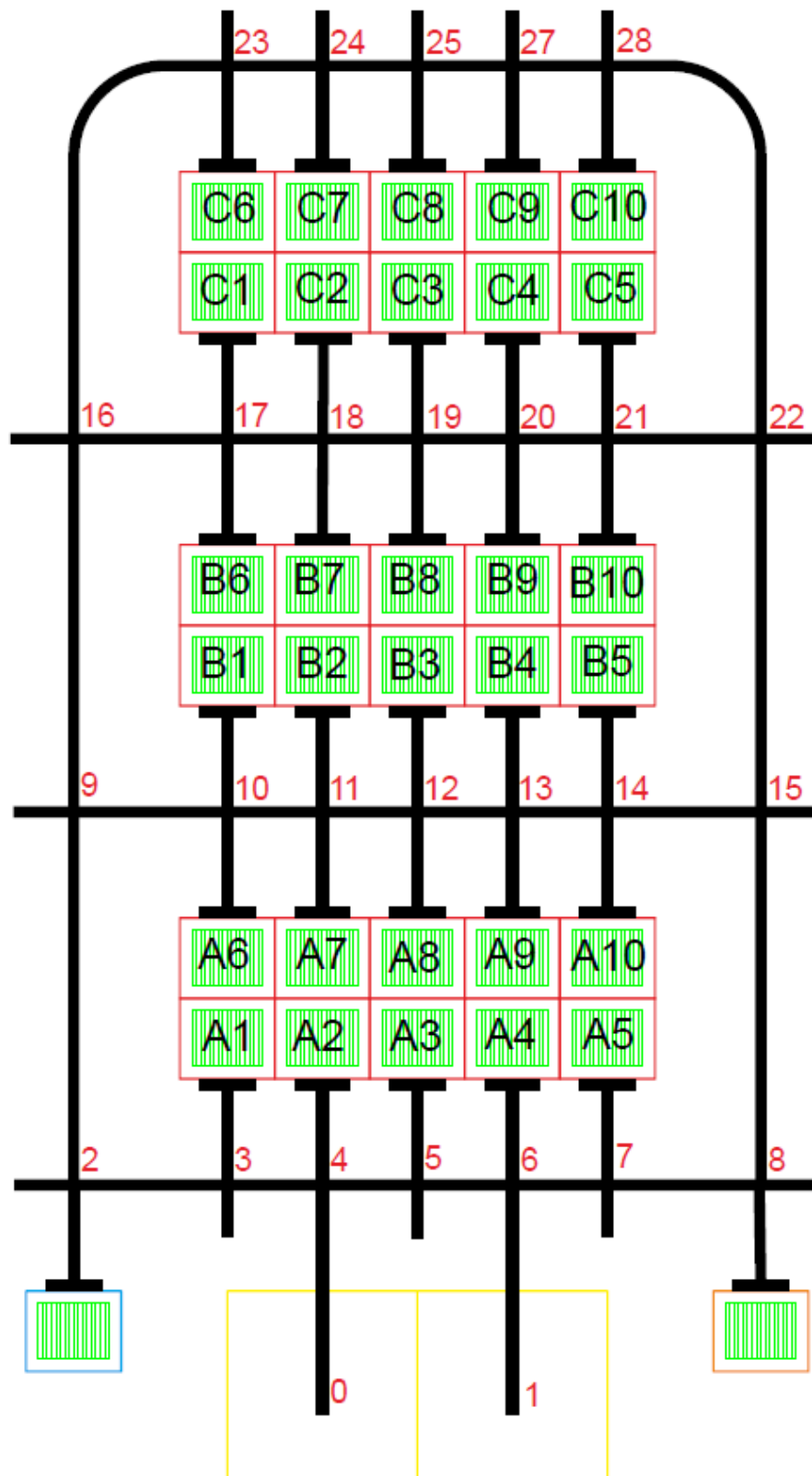
			<div>Relationship</div> <table><tr><td></td><td>Hubungan Kuat</td><td>8</td></tr><tr><td></td><td>Hubungan Moderat</td><td>5</td></tr><tr><td></td><td>Hubungan Lemah</td><td>1</td></tr></table>				Hubungan Kuat	8		Hubungan Moderat	5		Hubungan Lemah	1							
	Hubungan Kuat	8																			
	Hubungan Moderat	5																			
	Hubungan Lemah	1																			
			Costumer Importance	Model Dasar Forklift	Basis Pergerakan	Command & Data Processing Center	Transportasi Material	Charging and Standby Point													
Customer Requirements	Bentuk AGV	Ukuran AGV jangan terlalu besar	6																		
		Tipe AGV seperti <i>forklift</i> yang digunakan pada industri	4																		
	Tujuan Penggunaan	Mampu mensimulasikan kegiatan pergudangan yang mirip asli	8																		
		Memilih jalur terpendek saat bekerja	7																		
	Teknologi	Basis gerak yang umum dipakai AGV asli dan bisa diterapkan pada purwarupa	5																		
		Dikontrol dan bisa berkomunikasi dengan server secara <i>Wireless</i>	6																		
	Material Handling	Pengangkutan Material dengan Pallet	3																		
	Power Source and Charging Method	Pengisian Daya Baterai Secara Otomatis	4																		
			Absolute Importance (Priorities)	147	203	167	144	137													
			Relative Importance (%)	18.4	25.4	20.9	18.1	17.2													
		Target	Purwarupa AGV berbentuk seperti <i>forklift</i> aktual dan mampu beraktrifitas pada <i>aisle</i> yang sempit. Pergerakan berbasis perunut garis atau jalur Server memantau dan mengontrol purwarupa AGV, dengan jalur komunikasi RF yang datanya diolah oleh mikrokontroler Arduino Pengangkutan material menggunakan material palet Purwarupa AGV mampu kembali otomatis ke <i>charging point</i> saat daya mencapai batas minimum baterai																		
		Evaluasi Teknik (Benchmarking)	5																		
			4																		
			3																		
			2																		
			1																		

Correlation	
	Positif Kuat
	Positif
	Negatif
	Negatif kuat

Correlation	
	Positif Kuat
	Positif
	Negatif
	Negatif kuat



**LAMPIRAN 3**  
**LAYOUT GUDANG**



## LAMPIRAN 4

### PROGRAM MIKROKONTROLER

Program Mikrokontroler Arduino Nano 1:

```

/*
 * PINOUT PING SENSOR ==== FLEKSIBEL, PIN BISA DIRUBAH SESUKA
 HATI=====
 * -----
 * PING      ARDUINO NANO
 * -----
 * GND      GND
 * 5V      5V
 * TRIGGER   0
 * ECHO      1
 */
#include <Servo.h>           // LIBRARY SERVO
#include <NewPing.h>         // LIBRARY PING SENSOR

const int power = 100;       //NILAI SPEED KONSTAN NON PID
const int iniMotorPower = 250; //NILAI SPEED KONSTAN PID JALAN
NORMAL
const int iniMotorPower1 = 100; //NILAI SPEED KONSTAN PID JALAN PELAN
float adjTurn = 11.45;       //NILAI ADJUSTER DELAY BERBELOK

const int LFS0 = 14;         //DEFINE NAMA SENSOR INFRAMERAH DENGAN PIN
const int LFS1 = 15;
const int LFS2 = 16;
const int LFS3 = 17;
const int LFS4 = 18;

int LFSensor[5] = {0, 0, 0, 0, 0};

// PID controller
float Kp = 60;               // KONSTANTA PID JALAN NORMAL
float Ki = 0;
float Kd = 0;
float Kp1 = 20;              // KONSTANTA PID JALAN PELAN
float Ki1 = 0;
float Kd1 = 0;

float error = 0, P = 0, I = 0, D = 0, PIDvalue = 0; //SETTING AWAL VARIABEL PID 0
float error1 = 0, P1 = 0, I1 = 0, D1 = 0, PIDvalue1 = 0;
float previousError = 0, previousI = 0;
float previousError1 = 0, previousI1 = 0;
#define RIGHT 1
#define LEFT -1

#define LineFollow 8         // DEFINE ALAMAT TERIMA DARI NANO 1 PADA PIN
#define Parking 7
#define Pick 6
#define Drop 5
#define TR90 3
#define TL90 4
#define TL180 2

```

```
#define TRIGGER_PIN 0      // SETTING PIN TRIGGER DAN ECHO PING SENSOR
#define ECHO_PIN 1
#define MAX_DISTANCE 25    // SETTING JARAK DETEKSI PING (CM)
```

```
Servo leftServo;
Servo rightServo;
Servo forkServo;
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //INISIALISASI PING
DNG NAMA SONAR
```

```
//-----
/* NILAI ERROR PEMBACAAN SENSOR INFRA MERAH
```

Sensor Array	Error	
0 0 0 1	5	//UNTUK MEMPERTAJAM BELOKAN
0 0 0 1 1	4	
0 0 0 1 0	2	
0 0 1 1 0	1	
0 0 1 0 0	0	
0 1 1 0 0	-1	
0 1 0 0 0	-2	
1 1 0 0 0	-4	
1 0 0 0 0	-5	
0 0 0 0 0	0	ABAIKAN GARIS PENUH

```
*/
void readLFSensors()
{
  LFSensor[0] = digitalRead(LFS0);
  LFSensor[1] = digitalRead(LFS1);
  LFSensor[2] = digitalRead(LFS2);
  LFSensor[3] = digitalRead(LFS3);
  LFSensor[4] = digitalRead(LFS4);

  if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 1 )) {
    error = 5;
  }
  else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 1 )) {
    error = 4;
  }
  else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 0 )) {
    error = 2;
  }
  else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 0 )) {
    error = 1;
  }
  else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
    error = 0;
  }
  else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
    error = - 1;
  }
}
```

```

    }
    else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
        error = -2;
    }
    else if ((LFSensor[0] == 1 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
        error = -4;
    }
    else if ((LFSensor[0] == 1 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
        error = -5;
    }
    else if ((LFSensor[0] == 1 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 1 )) {
        error = 0;
    }
}

//-----
/* NILAI ERROR PEMBACAAN SENSOR INFRA MERAH

Sensor Array   Error
0 0 0 0 1      4
0 0 0 1 1      3
0 0 0 1 0      2
0 0 1 1 0      1
0 0 1 0 0      0
0 1 1 0 0     -1
0 1 0 0 0     -2
1 1 0 0 0     -3
1 0 0 0 0     -4
0 0 0 0 0

*/
void readLFSensors1()
{
    LFSensor[0] = digitalRead(LFS0);
    LFSensor[1] = digitalRead(LFS1);
    LFSensor[2] = digitalRead(LFS2);
    LFSensor[3] = digitalRead(LFS3);
    LFSensor[4] = digitalRead(LFS4);

    if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 1 )) {
        error1 = 4;
    }
    else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 1 )) {
        error1 = 3;
    }
    else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 0 )) {
        error1 = 2;
    }
    else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 0 )) {
        error1 = 1;
    }
}

```

```

}
else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
    error1 = 0;
}
else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
    error = - 1;
}
else if ((LFSensor[0] == 0 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
    error1 = -2;
}
else if ((LFSensor[0] == 1 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
    error1 = -3;
}
else if ((LFSensor[0] == 1 ) && (LFSensor[1] == 0 ) && (LFSensor[2] == 0 ) &&
(LFSensor[3] == 0 ) && (LFSensor[4] == 0 )) {
    error1 = -4;
}
else if ((LFSensor[0] == 1 ) && (LFSensor[1] == 1 ) && (LFSensor[2] == 1 ) &&
(LFSensor[3] == 1 ) && (LFSensor[4] == 1 )) {
    error1 = 0;
}
}

```

```

//-----
void motorStop()
{
    leftServo.writeMicroseconds(1500);
    rightServo.writeMicroseconds(1500);
    delay(300);
}

```

```

//-----
void motorForward()
{
    leftServo.writeMicroseconds(1500 - power);
    rightServo.writeMicroseconds(1500 + power);
}

```

```

//-----
void motorBackward()
{
    leftServo.writeMicroseconds(1500 + power);
    rightServo.writeMicroseconds(1500 - power);
}

```

```

//-----
void motorFwTime (unsigned int time)
{
    motorForward();
    delay (time);
}

```

```

    motorStop();
    delay(500);
}

//-----
void motorBwTime (unsigned int time)
{
    motorBackward();
    delay (time);
    motorStop();
    delay(500);
}

//-----
void motorTurn(int direction, int degrees)
{
    leftServo.writeMicroseconds(1500 - iniMotorPower * direction);
    rightServo.writeMicroseconds(1500 - iniMotorPower * direction);
    delay (round(adjTurn * degrees + 1));
    motorStop();
}

//-----
void motorPIDcontrolFWD()
{
    int leftMotorSpeed = 1500 - iniMotorPower - PIDvalue;
    int rightMotorSpeed = 1500 + iniMotorPower - PIDvalue;

    // The motor speed should not exceed the max PWM value
    constrain(leftMotorSpeed, 1000, 2000);
    constrain(rightMotorSpeed, 1000, 2000);

    leftServo.writeMicroseconds(leftMotorSpeed);
    rightServo.writeMicroseconds(rightMotorSpeed);
}

//-----
void motorPIDcontrolFWDP()
{
    int leftMotorSpeed = 1500 - iniMotorPower1 - PIDvalue1;
    int rightMotorSpeed = 1500 + iniMotorPower1 - PIDvalue1;

    // The motor speed should not exceed the max PWM value
    constrain(leftMotorSpeed, 1000, 2000);
    constrain(rightMotorSpeed, 1000, 2000);

    leftServo.writeMicroseconds(leftMotorSpeed);
    rightServo.writeMicroseconds(rightMotorSpeed);
}

//-----
void motorPIDcontrolBCK()
{

```

```

int leftMotorSpeed = 1500 + iniMotorPower1 + PIDvalue1 ;
int rightMotorSpeed = 1500 - iniMotorPower1 + PIDvalue1 ;

// The motor speed should not exceed the max PWM value
constrain(leftMotorSpeed, 1000, 2000);
constrain(rightMotorSpeed, 1000, 2000);

leftServo.writeMicroseconds(leftMotorSpeed);
rightServo.writeMicroseconds(rightMotorSpeed);
}

//-----
void stepBCK() //mode
{
  readLFSensors1();
  calculatePID1();
  motorPIDcontrolBCK();
}
void stepFWD()
{
  readLFSensors();
  calculatePID();
  motorPIDcontrolFWD();
}
void stepFWDP()
{
  readLFSensors1();
  calculatePID1();
  motorPIDcontrolFWDP();
}

//-----
void pickUp() //mode
{
  while ((digitalRead(LFS1) == 0 ) || (digitalRead(LFS2) == 0 ) || (digitalRead(LFS3) == 0
))
  {
    stepFWDP();
  }
  motorStop();
  delay(1000);
  for (int a = 170; a > 125; a=a-1) //Forklift Naik
  {
    forkServo.write(a);
    delay(50);
  }
  delay(1000);
  // motorBwTime(150);
  while (digitalRead(Pick) == LOW) //pin Take
  {
    // stepBCK();
    motorBackward();
  }
}

```

```

    motorStop();
}

//-----
void dropDown()
{
    while ((digitalRead(LFS1) == 0 ) || (digitalRead(LFS2) == 0 ) || (digitalRead(LFS3) == 0
))
    {
        stepFWDP();
    }
    motorStop();
    delay(1000);
    for (int a = 125; a < 170; a++) //Forklift Turun
    {
        forkServo.write(a);
        delay(50);
    }
    delay(1000);
    // motorBwTime(150);
    while (digitalRead(Drop) == LOW) //pin Take
    {
        // stepBCK();
        motorBackward();
    }
    motorStop();
}
//-----

void calculatePID()
{
    P = error;
    I = I + error;
    D = error - previousError;
    PIDvalue = (Kp * P) + (Ki * I) + (Kd * D);
    previousError = error;
}
//-----

void calculatePID1()
{
    P1 = error1;
    I1 = I1 + error1;
    D1 = error1 - previousError1;
    PIDvalue1 = (Kp1 * P1) + (Ki1 * I1) + (Kd1 * D1);
    previousError1 = error1;
}
//-----

void Ping()
{
    delay(50);
    sonar.ping_cm();
    while (sonar.ping_cm() >=2 && sonar.ping_cm() <=20)
    {
        motorStop();
        delay(1000);
    }
}

```



```
//-----
void setup()
{
  pinMode(LFS0, INPUT);
  pinMode(LFS1, INPUT);
  pinMode(LFS2, INPUT);
  pinMode(LFS3, INPUT);
  pinMode(LFS4, INPUT);
  pinMode(LineFollow, INPUT);
  pinMode(Parking, INPUT);
  pinMode(Pick, INPUT);
  pinMode(Drop, INPUT);
  pinMode(TR90, INPUT);
  pinMode(TL90, INPUT);
  pinMode(TL180, INPUT);
  digitalWrite(LineFollow, HIGH);
  digitalWrite(Parking, HIGH);
  digitalWrite(Pick, HIGH);
  digitalWrite(Drop, HIGH);
  digitalWrite(TR90, HIGH);
  digitalWrite(TL90, HIGH);
  digitalWrite(TL180, HIGH);
  leftServo.attach(11);
  rightServo.attach(9);
  forkServo.attach(10);
  forkServo.write(170);
}

//-----
void loop()
{
  if (digitalRead(LineFollow) == LOW) //Jika Menerima Perintah LineFollow
  {
    while (digitalRead(LineFollow) == LOW)
    {
      stepFWD();
      Ping();
    }
    motorStop();
  }
  else if (digitalRead(Parking) == LOW) //Jika Menerima Perintah Parking
  {
    motorBwTime(500);
    delay(100);
    while (digitalRead(Parking) == LOW)
    {
      stepBCK();
    }
    motorStop();
  }

  else if (digitalRead(Pick) == LOW) //Jika Menerima Perintah Pick
```

```

{
while (digitalRead(Pick) == LOW)
{
    pickUp();
}
}
else if (digitalRead(Drop) == LOW) //Jika Menerima Perintah Drop
{
    while (digitalRead(Drop) == LOW)
    {
        dropDown();
    }
}
else if (digitalRead(TR90) == LOW) //Jika Menerima Perintah Putar Kanan 90
{
    motorStop();
    motorTurn(RIGHT, 92);
    previousError = 0;
    while(digitalRead(TR90) == LOW)
    {
    }
}
else if (digitalRead(TL90) == LOW) //Jika Menerima Perintah Putar Kiri 90
{
    motorStop();
    motorTurn(LEFT, 92);
    previousError = 0;
    while(digitalRead(TL90) == LOW)
    {
    }
}
else if (digitalRead(TL180) == LOW) //Jika Menerima Perintah Putar 180
{
    motorStop();
    motorTurn(LEFT, 180);
    previousError = 0;
    while(digitalRead(TL180) == LOW)
    {
    }
}
}
}

```

Program Arduino Nano 2:

```

/*
 * PINOUT RFID MFRC522 =====WARNING: VOLTAGE 3.3V, BUKAN 5V=====
 * -----
 *      MFRC522      Arduino
 *      Reader/PCD   Nano
 * Signal   Pin      Pin
 * -----
 * RST/Reset RST      D9
 * SPI SS    SDA(SS)  D10
 * SPI MOSI  MOSI     D11
 * SPI MISO  MISO     D12
 * SPI SCK   SCK      D13
 */

/*
 * PINOUT RF APC220 ==== FLEKSIBEL, PIN BISA DIRUBAH SESUKA HATI=====
 * -----
 * RF APC220  ARDUINO NANO
 * -----
 * GND        GND
 * 5V         5V
 * EN         NC
 * RX         8
 * TX         7
 * AUX        NC
 * SET        NC
 */

#include <stdio.h>           //LIBRARY EEPROM
#include <string.h>          //LIBRARY EEPROM
#include <EEPROM.h>          //LIBRARY EEPROM
#include <SPI.h>             // LIBRARY RFID
#include <MFRC522.h>         // LIBRARY RFID
#include <SoftwareSerial.h>  //LIBRARY RF APC220

const int pinRX = 7 ;       //SETTING PIN RX TX UNTUK RF APC220
const int pinTX = 8;

#define LineFollow 1        // DEFINE PIN PERINTAH KE NANO 2
#define Parking 0
#define Pick 2
#define Drop 3
#define TR90 4
#define TL90 5
#define TL180 6
#define SS_PIN 10           // DEFINE PIN RFID
#define RST_PIN 9

boolean match = false;     // INILIASASI AWAL MATCH KARTU RFID = SALAH

uint8_t successRead;       // VARIABEL UNTUK MENYIMPAN PEMBACAAN KARTU
RFID SUKSES
int alamat;

byte storedCard[4];        // VARIABEL SIMPANAN ALAMAT KARTU RFID DI
EEPROM
byte readCard[4];          // VARIABEL SIMPANAN PEMBACAAN KARTU RFID

```

```

String readString;

String titikRFIDtujuan;    // VARIABEL SIMPANAN PERINTAH TUJUAN DARI
SERVER
String modegerakan;        // VARIABEL SIMPANAN PERINTAH GERAKAN DARI
SERVER
int nilaibaterai;          // VARIABEL SIMPANAN NILAI BATERAI DARI
SERVER
String datakirimRFID = ""; // VARIABEL SIMPANAN SEND PERINTAH POSISI KE
SERVER
String databaterai = "";  // VARIABEL SIMPANAN SEND NILAI BATERAI KE
SERVER
String titikdua = ":";    // VARIABEL PEMBATAS ANTAR DATA KIRIMAN KE
SERVER

SoftwareSerial apc220(pinRX, pinTX); //SETTING FUNGSI SERIAL RF APC220
MFRC522 mfrc522(SS_PIN, RST_PIN);   //SETTING FUNGSI RFID

////////////////////// Setup ////////////////////////////////////////
void setup()
{
    pinMode(A3, INPUT);           //SETTING INPUT OUTPUT PIN
    pinMode(A2, INPUT);
    pinMode(LineFollow, OUTPUT);
    pinMode(Parking, OUTPUT);
    pinMode(Pick, OUTPUT);
    pinMode(Drop, OUTPUT);
    pinMode(TR90, OUTPUT);
    pinMode(TL90, OUTPUT);
    pinMode(TL180, OUTPUT);
    digitalWrite(LineFollow, HIGH);
    digitalWrite(Parking, HIGH);
    digitalWrite(Pick, HIGH);
    digitalWrite(Drop, HIGH);
    digitalWrite(TR90, HIGH);
    digitalWrite(TL90, HIGH);
    digitalWrite(TL180, HIGH);

    SPI.begin();                  // INISIALISASI PROTOKOL SPI UNTUK RFID
    mfrc522.PCD_Init();           // INISIALISASI RFID
    apc220.begin(9600);           // INISIALISASI RF APC220
    mfrc522.PCD_SetAntennaGain(mfrc522.RxGain_23dB); //SETTING GAIN RANGE
    RFID 23,33,38,43,48

    cekulangawal:
    while (1)
    {
        if (apc220.available()) //KETIKA KARAKTER MASUK KE KOMUNIKASI RF APC220
        EX PERINTAH: XX:XX:XX
        {
            while (apc220.available() > 0)
            {
                char c = apc220.read();
                readString += c;
            }
            readString.trim();      //potong kelebihan spasi atau enter di awal dan
            di akhir

```

```

        int ttkdua1 = readString.indexOf(":");           //menemukan lokasi/index titik dua (:)
yang pertama
        int ttkdua2 = readString.indexOf(":", ttkdua1+1); //menemukan lokasi/index titik dua
(:) yang kedua
        String C1= readString.substring(0,ttkdua1);      //perintah disimpan dalam variabel
C1, C2, C3
        String C2= readString.substring(ttkdua1+1,ttkdua2);
        String C3= readString.substring(ttkdua2+1);

        if (C1 == "01")                                //Jika perintah untuk agv 1, data kemudian
dipecah dan disimpan untuk dipakai
        {
            if (C3 == "S")
            {
                //Jika kiriman pertama di C3 bernilai S, cek status
                untuk pertama kali
                nilaibaterai = ceknilaibaterai();
                databaterai = String(nilaibaterai);
                apc220.print("1:0:0:500");
            }
            else
            {
                goto cekulangawal;
            }
        }
        else
        {
            goto cekulangawal; //mengulang terus sampai dapat perintah
            kiriman dari server data pertama 01
        }
        readString="";
        goto lanjut;
    }
    lanjut;;
}

//////////////////// Main Loop //////////////////////
void loop ()
{
    //menunggu ada data dari APC220

    ceklagi:
    if (apc220.available())
    {
        while (apc220.available()>0)
        {
            //memecah perintah kiriman server kemudian membandingkan isi
            perintah
            char c = apc220.read();
            readString += c;
        }
        Serial.println(readString);
        readString.trim();
        int ttkdua1 = readString.indexOf(":");

```

```

int ttkdua2 = readString.indexOf(":", ttkdua1+1);
String C1= readString.substring(0,ttkdua1);
String C2= readString.substring(ttkdua1+1,ttkdua2);
String C3= readString.substring(ttkdua2+1);

Serial.println(C1);
if (C1 == "1")
{
    titikRFIDtujuan = C2;
    modegerakan = C3;

    if (modegerakan == "MA")
server
    {
        //gerakan maju
        digitalWrite(LineFollow , LOW);
        LineFollow ke Nano2
        delay(1000);
    }
    else if (modegerakan == "MU")
dari server
    {
        //gerakan mundur
        digitalWrite(Parking , LOW);
        ke Nano2
        delay(1000);
    }
    else if (modegerakan == "KA")
server
    {
        //gerakan putar kanan 90
        digitalWrite(TR90 , LOW);
        Nano2 (belok kanan)
        delay(2000);
        digitalWrite(TR90 , HIGH);
        nilaibaterai = ceknilaibaterai();
        databaterai = String(nilaibaterai);
        datakirimRFID = String(alamat);
        apc220.print("1:0:" + datakirimRFID + ":" + databaterai);
        readString = "";
        goto ceklagi;
    }
    else if (modegerakan == "KI")
server
    {
        //gerakan putar kiri 90
        digitalWrite(TL90 , LOW);
        Nano2 (belok kiri)
        delay(2000); //delay waktu belok
        digitalWrite(TL90 , HIGH);
        nilaibaterai = ceknilaibaterai();
        databaterai = String(nilaibaterai);
        datakirimRFID = String(alamat);
        apc220.print("1:0:" + datakirimRFID + ":" + databaterai);
        readString = "";
        goto ceklagi;
    }
}

```

//Jika menerima perintah "MA" dari server

//Mengaktifkan perintah Pin LineFollow ke Nano2

//Jika menerima perintah "MU" dari server

//Mengaktifkan perintah Pin Parking ke Nano2

//Jika menerima perintah "KA" dari server

//Mengaktifkan perintah Pin TR90 ke Nano2 (belok kanan)

// cek dan kirim nilai baterai

//Jika menerima perintah "KI" dari server

//Mengaktifkan perintah Pin TL90 ke Nano2 (belok kiri)

```

else if (modegerakan == "AN") //Jika menerima perintah "AN"
dari server
{
    //gerakan angkat
    digitalWrite(Pick , LOW); //Mengaktifkan perintah Pin Pick ke
Nano2
    delay(2000);
}
else if (modegerakan == "TU") //Jika menerima perintah "TU"
dari server
{
    //gerakan nurunin
    digitalWrite(Drop , LOW); //Mengaktifkan perintah Pin Drop ke
Nano2
    delay(2000);
}
else if (modegerakan == "PB") //Jika menerima perintah "PB"
dari server
{
    //gerakan memutar 180
    digitalWrite(TL180 , LOW); //Mengaktifkan perintah Pin TL180
ke Nano2 (memutar)
    delay(4000); //delay waktu belok
    digitalWrite(TL180 , HIGH);
    nilaibaterai = ceknilaibaterai();
    databaterai = String(nilaibaterai);
    data kirimRFID = String(alamat);
    apc220.print("1:0:" + data kirimRFID + ":" + databaterai);
    readString = "";
    goto ceklagi;
}
else if (modegerakan == "S")
{
    //nanya status
    nilaibaterai = ceknilaibaterai();
    databaterai = String(nilaibaterai);
    apc220.print("1:0:0:" + databaterai);
    readString="";
    delay(2000);
    goto ceklagi;
}
else
{
    //do nothing
}

cekulangkartu:
//sekarang cek kartu
successRead = false;
while (!successRead) //HOLDING BILA KARTU POSISI SALAH / TIDAK
TERDAFTAR
{
    successRead = getID();
}
if (findID(readCard) == true )
{
    //jika kartu yang dibaca sama dengan data di eeprom
    if (String(alamat) == titikRFIDtujuan)

```

```

{
  //jika alamat yang didapat sama dengan C2
  digitalWrite(LineFollow , HIGH);
  digitalWrite(Parking , HIGH);
  digitalWrite(TL90 , HIGH);
  digitalWrite(TR90 , HIGH);
  digitalWrite(TL180 , HIGH);
  digitalWrite(Pick , HIGH);
  digitalWrite(Drop , HIGH);          // mematikan semua pin alamat

                                  //mengirim data ke server
  nilaibaterai = ceknilaibaterai();
  databaterai = String(nilaibaterai);
  datakirimRFID = String(alamat);
  apc220.print("1:0:" + datakirimRFID + ":" + databaterai);
  successRead = false;
}
else
{
  mfr522.PICC_HaltA();          // RFID Stop reading jika beda dengan
  tujuan kiriman dari server
  goto cekulangkartu;
}
}
else
{
  //jika kartu berbeda, kembali ke atas, baca ulang kartunya
  goto cekulangkartu;
}
}
else
{
  readString="";
  goto ceklagi; //mengulang terus sampai dapat perintah kiriman dari server data
  pertama 01
}
readString="";
}
}

/////////////////////// Cek nilai baterai terkecil /////////////////////////
int ceknilaibaterai()
{
  if (analogRead(A3) <= analogRead(A2))
  {
    return analogRead(A3);
  }
  else
  {
    return analogRead(A2);
  }
}

/////////////////////// Mendapatkan bacaan alamat RFID card///////////////////////
uint8_t getID()
{
  // Getting ready for Reading PICCs

```



```

    if ( ! mfrc522.PICC_IsNewCardPresent()) { //If a new PICC placed to RFID reader
continue
        return 0;
    }
    if ( ! mfrc522.PICC_ReadCardSerial()) { //Since a PICC placed get Serial and continue
        return 0;
    }
    // There are Mifare PICCs which have 4 byte or 7 byte UID care if you use 7 byte PICC
    // I think we should assume every PICC as they have 4 byte UID
    // Until we support 7 byte PICCs

    //Serial.println(F("Scanned PICC's UID:"));
    for ( uint8_t i = 0; i < 4; i++) { //
        readCard[i] = mfrc522.uid.uidByte[i];
        //Serial.print(readCard[i], HEX);
    }
    //Serial.println("");
    // mfrc522.PICC_HaltA(); // Stop reading
    return 1;
}

//////////////////// Membaca ID dari EEPROM //////////////////////
void readID( uint8_t number )
{
    uint8_t start = (number * 5) + 1; // Figure out starting position //data kartu mulainya
    dari 6, 11, 16, 21, dst
    for ( uint8_t i = 0; i < 4; i++) { // Loop 4 times to get the 4 Bytes
        storedCard[i] = EEPROM.read(start + i); // Assign values read from EEPROM to array
    }
}

//////////////////// Check Bytes //////////////////////
boolean checkTwo ( byte a[], byte b[] )
{
    if ( a[0] != 0 ) // Make sure there is something in the array first
    {
        match = true; // Assume they match at first
        for ( uint8_t k = 0; k < 4; k++ ) // Loop 4 times
        {
            if ( a[k] != b[k] ) // IF a != b then set match = false, one fails, all fail
            {
                match = false;
            }
        }
    }
}

if ( match ) // Check to see if if match is still true
{
    return true; // Return true
}
else
{
    return false; // Return false
}
}

//////////////////// Menemukan ID dari EEPROM //////////////////////
boolean findID( byte find[] ) {

```

```

uint8_t count = EEPROM.read(0); // Read the first Byte of EEPROM that
for ( uint8_t i = 1; i <= count; i++ ) { // Loop once for each EEPROM entry
  readID(i); // Read an ID from EEPROM, it is stored in storedCard[4]

// dapat data kartu dengan index i di eeprom, data kartu dari eeprom disimpan di
storedcard[4]
  if ( checkTwo( find, storedCard ) )
  { // Check to see if the storedCard read from EEPROM // kalo sama dengan data
    yang di eeprom, masukan data berikutnya ke alamat.
    alamat = EEPROM.read((i*5)+5);
    return true;
    break; // Stop looking we found it
  }
  else { // If not, return false
  }
}
return false;
}

```



**LAMPIRAN 5**  
**FORM UJI COBA KINERJA PURWARUPA AGV**

1. Penyimpanan / Pengambilan

No.	Sel	Tes 1	Tes 2	Tes 3	Tes 4	Tes 5	Jumlah
1	A1						
2	A2						
3	A3						
4	A4						
5	A5						
6	A6						
7	A7						
8	A8						
9	A9						
10	A10						
11	B1						
12	B2						
13	B3						
14	B4						
15	B5						
16	B6						
17	B7						
18	B8						
19	B9						
20	B10						
21	C1						
22	C2						
23	C3						
24	C4						
25	C5						
26	C6						
27	C7						
28	C8						
29	C9						
30	C10						
Rata-rata							
Presentase							

## 2. Kembali ke *Docking Station*

No.	RFID Awal	Tes 1	Tes 2	Tes 3	Tes 4	Tes 5	Jumlah
1	2						
2	3						
3	4						
4	5						
5	6						
6	7						
7	8						
8	10						
9	11						
10	12						
11	13						
12	14						
13	17						
14	18						
15	19						
16	20						
17	21						
18	23						
19	24						
20	25						
21	26						
22	27						
Rata-rata							
Presentase							

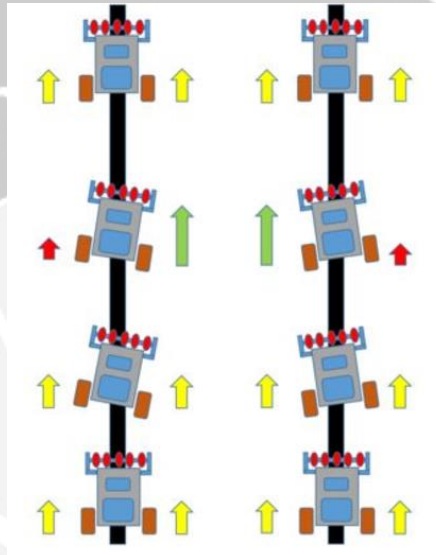
## 3. *Feedback*

No.	Feedback	Tes 1	Tes 2	Tes 3	Tes 4	Tes 5	Jumlah
1	Posisi (random)						
2	Status <i>on duty</i>						
3	Selesai bertugas						
4	Status Baterai						
Rata-rata							
Presentase							

## LAMPIRAN 6

### PERANCANGAN PENGEMBANGAN MATERI PEMBELAJARAN

1. Gambarkan dan jelaskan cara kerja dan keterkaitan komponen pada purwarupa AGV dengan modul yang sudah dipelajari di praktikum.
2. Jelaskan cara kerja sensor infra merah dan motor sehingga bisa bergerak merunut garis! Petunjuk bisa dilihat pada gambar berikut.



3. Bagaimana prinsip kerja Sensor Ping dalam mendeteksi halangan yang ada di depannya?
4. Dalam kalkulasi PID kontrol Proporsional pada AGV, *error* digunakan sebagai pembenahan gerakan / *output*. Rumus PID yang digunakan dalam program Arduino adalah:

$$PID = (Kp * P) + (Ki * I) + (Kd * D)$$

dengan

$$P = error, \quad I = I + error, \quad D = error + previous\ error$$

Bila menggunakan kontrol Proporsional, maka yang memiliki nilai adalah Kp, dengan permissalan nilai konstan 50. Apabila kecepatan konstan motor adalah 250 dan penerjemahan *error* dari sensor infra merah saat membaca garis adalah sebagai berikut

Sensor Array	Error
0 0 0 0 1	4
0 0 0 1 1	3
0 0 0 1 0	2
0 0 1 1 0	1
0 0 1 0 0	0
0 1 1 0 0	-1
0 1 0 0 0	-2
1 1 0 0 0	-3
1 0 0 0 0	-4

Hitunglah kecepatan motor saat *error*.

Motor/Error	4	3	2	1	0	-1	-2	-3	-4
Motor Kanan									
Motor Kiri									

- Gambarkan kondisi *error* dan kecepatan pembenahan kondisi AGV dengan ilustrasi seperti nomor 3.
- Berikan pendapat anda mengenai simulasi yang dilakukan oleh purwarupa AGV, dan seberapa jauh pengaruhnya dalam membantu pemahaman anda dalam sistem otomasi dan integrasi komponen sistem kendali.

## LAMPIRAN 7

### TRANSKRIP WAWANCARA DAN *CUSTOMER REQUIREMENTS*

1. Pada praktikum SKI, mahasiswa belum bisa memahami fungsi dan kapabilitas penggunaan PLC dan mikrokontroler dalam sistem otomasi, apakah hal ini benar adanya menurut dosen?

**Jawaban:** Hal tersebut memang benar, tidak semua siswa mampu memahami penggunaan komponen tersebut dalam sistem otomasi di industri.

2. Pada praktikum SKI, mahasiswa belum bisa memahami fungsi, cara kerja, dan integrasi sensor pada kontroler yang ada, dan penerapannya dalam sistem otomasi, apakah hal ini benar adanya menurut dosen?

**Jawaban:** Hal itu juga benar, dalam pembuatan tugas besar dari tahun ke tahun mengenai otomasi, masalah peletakan sensor, jenis, dan spesifikasi kerja menjadi masalah tersendiri.

3. Pada mata kuliah yang diajarkan di perkuliahan teori, dalam hal ini SKI, Fisika, Mekatronika, dan juga Otomasi Industri, apakah mahasiswa sudah pernah diajarkan mengenai komponen kerja otomasi, sistem, dan cara kerja, serta bagaimana pemahaman mereka mengenai hal tersebut dalam praktikum yang dilakukan?

**Jawaban:** Mahasiswa sudah mempelajari hal-hal yang berkaitan dengan otomasi tersebut, baik komponen ataupun sistem kerjanya. Untuk bisa mengambil praktikum SKI, tentu mata kuliah tersebut juga harus lulus, tapi nyatanya apa yang sudah diajarkan, mahasiswa tidak bisa menganalisa keterkaitannya dengan apa yang ada di praktikum, seperti contoh perbedaan sensor digital dan analog, cara kerja *relay*, dan lainnya.

4. Dalam pembelajaran di praktikum SKI, materi yang diajarkan terkesan individu per komponen, dan mahasiswa tidak bisa memahami keterkaitan satu modul dengan yang lainnya, bagaimana pendapat Dosen mengenai hal ini dan apa sarannya?

**Jawaban:** Praktikum SKI setidaknya harus ada 1 modul yang menunjukkan gabungan materi-materi yang diajarkan, menjadi 1 alat peraga, ada komponen input seperti sensor, kontroler, dan komponen output. Jika memungkinkan berkaitan dengan otomasi pergudangan.

*Brainstorming* konsep purwarupa:

1. Otomasi di pergudangan yang digunakan salah satunya adalah *Automated Guided Vehicle (AGV)*, bagaimana menurut dosen apabila digunakan sebagai konsep alat peraga?

**Jawaban:** Boleh saja, apabila menggunakan AGV sekaligus menunjukkan jalur terpendek dan ditunjukkan pada antarmuka komputer. AGV berbentuk seperti *forklift* pada umumnya yang digunakan di industri.

2. Fungsi dan kegiatan apa saja yang mau disimulasikan dengan dengan AGV?

**Jawaban:** Tentu kegiatan umum yang biasa dilakukan dalam pergudangan seperti simpan dan ambil, dan lainnya. Dan juga harus ada *material handling* dalam proses kerjanya.

3. Fitur apa yang kira-kira mau ditambahkan dalam purwarupa ini?

**Jawaban:** Yang jelas bisa dipantau oleh komputer atau antar muka. Lalu cara komunikasinya bagaimana? *Wireless*? Atau apa? Penggunaan sumber daya mampu mengisi ulang sendiri atau kembali saat batas minimal pemakaian ke titik *standby*. Semua status kesiapan dan proses kerja pokoknya dipantau oleh server.

4. Komponen yang digunakan untuk pembuatan purwarupa apakah boleh diluar modul atau materi yang diajarkan di praktikum SKI?

**Jawaban:** Sebisa mungkin dimaksimalkan dengan komponen dari materi pembelajaran, apabila memang dibutuhkan baru memakai komponen yang diluar itu.



No	Customer Requirements	Bobot	Keterangan
1	Ukuran AGV jangan terlalu besar	6	Ukuran jangan terlalu besar, menyesuaikan luas lab dan <i>layout</i> .
2	Tipe AGV seperti forklift yang biasa untuk industri	4	Bentuknya sebisa mungkin mirip dengan forklift yang biasanya
3	Bisa mensimulasikan kegiatan gudang yang mirip asli	8	Purwarupa harus dibuat agar bisa menunjukkan simulasi kegiatan pergudangan.
4	Memilih jalur terpendek saat bekerja	7	Setiap proses kerja AGV harus bisa memilih jalur terpendek.
5	Basis gerak yang umum dipakai di AGV asli dan bisa dipakai di purwarupa	5	Basis gerak menyesuaikan kondisi komponen dan area yang tersedia, namun sebisa mungkin sama dengan asli
6	Dikontrol dan bisa berkomunikasi dengan server secara <i>Wireless</i>	6	AGV nantinya dipantau dengan server untuk status kerja dan kegiatannya
7	Pengangkutan material dengan palet	3	Material / model benda yang diangkut dengan media palet.
8	Pengisian Daya Baterai Secara Otomatis	4	Jika memungkinkan, saat baterai habis / memasuki batas minimal, bisa mengisi sendiri.

Range pembobotan adalah 1-9, dengan kriteria sebagai berikut:

Bobot Prioritas Konsumen	Keterangan
1	Sangat Tidak Penting Sekali
2	Sangat Tidak Penting
3	Tidak Penting
4	Agak Tidak Penting
5	Netral
6	Agak Penting
7	Penting
8	Sangat Penting
9	Sangat Penting Sekali